FACSIMILE

DATE: **May 11, 2009**   JOB CODE:

OFFICIAL PAPER

Please deliver this and the following pages to:

Examiner:	**Benjamin R. Bruckart**
U.S.P.T.O. Group Art Unit:	**2446**
Telecopier No.:	**Sent via email: Benjamin.Bruckart@uspto.gov**
U.S. Serial No.:	**10/609,426**
Client/Matter No.:	****BU-0124**
Sender's Name:	**Aaron F. Bourgeois**
Pages to Follow:	**23**

If transmission is not complete, please call **Aaron F. Bourgeois** at **(404) 459-5522**.

COVER MESSAGE:

OFFICIAL FACSIMILE. PLEASE DELIVER TO EXAMINER IMMEDIATELY.

Please find attached a proposed agenda for the interview regarding the above-listed application scheduled for **Tuesday, May 11, 2009 at 11:00 AM**.

Sincerely,

/Aaron F. Bourgeois/
Aaron F. Bourgeois
Registration No. 57,936

WOODCOCK WASHBURN

INTELLECTUAL PROPERTY LAW

ATLANTA • PHILADELPHIA • SEATTLE

FACSIMILE

Proposed Agenda for Examiner Interview

- Discuss the proposed claim amendments (example amended claim is provided below), and the application of the cited references (Mukherjee (6,466,978) and Bestavros (6,370,584)). Note that I have attached a portion of a draft version of the office action response containing the amended claims, drawings and specification for your ease of reference since the amendments draw at least in part from material amended into the specification from an incorporated application. Specifically:

  o The amended independent claims include the elements attempting to capture the concept of storing content on an adaptable cache of a server in a group of servers and transmitting this information to other servers in the group. The closest cited art in the view of applicants is Mukherjee as cited in the office action (col. 15, line 49 – col. 16, line 14), which describes adjusting the size of a cluster by looking at the loads of servers in the cluster and adding/subtracting servers to the cluster based on load. Mukherjee does not appear to disclose or suggest adaptable caches and using them as claimed. Bestavros also fails to disclose or suggest this aspect.

  o The amended independent claims also include the elements of determining that a director server's adaptable cache does not have the content, and examining parametric information about the adaptable caches of other servers in the plurality of servers to select a server to service a request. The reference do not appear to disclose this aspect.

  o In the dependent claims, the aspect of storing the requested content on a director's adaptable cache responsive to a request for the content is claimed. Applicants also do not see this aspect in the cited art.

FACSIMILE

Amendments to the Specification:

Please replace paragraph [0048] in its entirety with the following paragraph, wherein markings are included to show changes made.

[0048]   In a preferred embodiment, one or more of the stored parameters relate to the asset inventory of each server. The state table may also store other media asset parameters such as whether the asset is a "new release" to help anticipate demand for the asset. The state table additionally may contain parameters concerning the capability of each server such as whether it comprises extended memory or an inline adaptable cache (such as that described in U.S. patent application serial No. **10/609,433, now U.S. patent No. 7,500,055**, entitled **"ADAPTABLE CACHE FOR DYNAMIC DIGITAL MEDIA"**, filed **June 27,2003,** ~~(identified by Pennie & Edmonds docket No. 11055-013)~~ which is hereby incorporated by reference in its entirety for each of its teachings and embodiments), or other unique storage attributes.

Please add the following paragraphs to the specification, between paragraphs [0048] and [0049]. The paragraphs presented below are adapted from paragraphs [0080]-[0109] of U.S. Patent Application No. 10/609,433, now U.S. Patent No. 7,500,055, which was incorporated by reference in its entirety at the time of filing the present application. No new matter is added.

FACSIMILE

[0048A]   In a preferred embodiment, an adaptable cache is adapted to proactively cache resources, and is further adapted to notify potential calling applications and other processes of assets it maintains.

[0048B]   Alternatively or in addition, the adaptable cache may be adapted to direct the storage system not to respond to requests for particular assets when the assets are cached in the adaptable cache. Operation of one preferred embodiment for implementing proactive caching and notification is described in connection with Fig. 9.

[0048C]   As shown in Fig. 9, in step 901, an adaptable cache monitors an I/O bus for asset requests. These may represent requests for content to be delivered immediately or requests for content to be delivered at a specified later time.

[0048D]   When a request is detected, the adaptable cache determines whether a copy of some or all of the asset is stored in a storage medium (step 902). In step 903, the adaptable cache further evaluates the request in accordance with one or more caching rules programmed into a core logic. In a preferred embodiment, these caching rules may take account of parameters maintained by the core logic, such as available capacity in the adaptable cache and the request frequency for the requested asset.

[0048E]   On the basis of steps 902-903, the adaptable cache determines whether or not some or all of the requested asset or some related asset should be proactively cached (step 904). If it is determined that some or all of an asset should be proactively cached, the system proceeds to step

905 where the adaptable cache communicates directly with the appropriate storage system or device and transfers all or a portion of the asset into its storage medium.

[0048F]   In step 906, the adaptable cache notifies requesting applications and other processes that may require the requested asset of its updated content so that future requests for that asset may be directed to the adaptable cache. These applications/processes, or associated hardware or software may preferably maintain a table that lists assets available from the adaptable cache. Each entity receiving notification from the adaptable cache preferably updates its table appropriately to reflect the current content of the adaptable cache. Processing then proceeds to step 907, described below.

[0048G]   If in step 904 it is determined not to cache requested content, the system proceeds directly to step 907 where parameters maintained by the core logic are updated. In a preferred embodiment, such parameters may, for example, include the number of times a particular asset has been requested within a specified amount of time and available capacity within the adaptable cache. Processing then returns to step 901 where the adaptable cache continues to monitor the I/O bus.

[0048H]   As will be recognized by those skilled in the art, passive monitoring of a bus by an adaptable cache as described above may be impractical with more modern busses which are often segmented and behave more like networks in which each device sees only traffic specifically addressed to it. Accordingly, in systems comprising such busses, a network interface may be adapted to address each received asset request to both a host processor and to an

adaptable cache so that the adaptable cache may monitor traffic between the network interface and the host processor. References to monitoring by the adaptable cache herein should be understood to include both passive monitoring as well as monitoring using such a dual addressing scheme.

[0048I]   Alternatively or in addition, an adaptable cache may be adapted to perform interval caching wherein a sorted list of pairs of overlapping requests for the same asset is maintained that identifies pairs of requests with the shortest intervals between their start times. For these pairs, as the first request in the pair is streamed, the streamed content is also cached and then read from cache to serve the second request.

[0048J]   One preferred embodiment for operation of a media server comprising an adaptable cache adapted for proactive caching and notification will now be described in connection with Fig. 10. As shown in Fig. 10, in step 1001, a request for an asset is received via a network interface and forwarded to a host processor via an I/O bus. In step 1002, the adaptable cache monitors the I/O bus for such requests, caches appropriate content if warranted under its caching rules, and notifies any requesting applications (including the requesting application running on host processor) of its updated content, as described above in connection with Fig. 9.

[0048K]   In step 1003, the host processor determines whether or not the requested asset is available from the adaptable cache, such as by consulting a table that stores current assets maintained by the adaptable cache. If the asset (or some portion of the asset) is available from the adaptable cache, the host processor formulates a request for the asset (or portion thereof) to

the adaptable cache (step 1004). In step 1005, the adaptable cache returns the requested asset to the host processor.

[0048L]   Otherwise, if the asset is not available from the adaptable cache, the host processor formulates a request for the asset to a storage system (step 1006). The requested asset is read in blocks from a storage device of the storage system and transmitted to the host processor as shown by the iteration of steps 1007-1010. More particularly, for each block, the storage device finds the block on the hard drive (step 1007), reads the block (step 1008), transmits the block (step 1009), and determines whether or not the asset comprises additional blocks (step 1010).

[0048M]   Another preferred embodiment for implementing the present system and method is shown in connection with Fig. 11. As in Fig. 10, adaptable cache 600 in Fig. 11 also resides as a device connected to the host side I/O bus 106B. In this embodiment, however, adaptable cache 600 is preferably integrated with network interface 130. The adaptable cache 600 preferably interconnects with the host side I/O bus 106B via interface connection 146. Preferred physical specifications for the adaptable cache in this preferred embodiment comprise:

- the form factor of a network interface card (e.g., a peripheral component interconnect or PC1 card) which may be plugged into an available expansion slot on the host system (e-g., a PC1 slot);

- storage capacity in excess of 1 gigabyte (GB) using replaceable commercially off-the-shelf memory modules ,(such as dual inline memory modules -DIMMs) or fixed memory circuits; and

- conformity to PC1 hot-swap specifications to allow the adaptable cache to be removed from service while the host system is in operation. As noted above, the storage size of the adaptable cache can therefore be altered through a hot-swap without disrupting the operation of the media server.

[0048N]   In this preferred embodiment, adaptable cache 600 is programmed to respond directly to asset requests when the requested asset is available in its storage medium. In this way, asset requests may be serviced and delivered from the network interface card, eliminating bus traversals when assets requested by the user reside in the adaptable cache.

[0048O]   Operation of the system shown in Fig. 11 will now be described in connection with Fig. 12. In step 1201, an asset request is received at network interface 130. In step 1202, adaptable cache 600 determines if the requested asset is available on the adaptable cache.

[0048P]   If the asset is available on the adaptable cache, the request is preferably serviced and delivered to the user from the same card, eliminating bus traversals on buses 106 (step 1203). More specifically, the adaptable cache retrieves the resource from its storage medium, converts it to an appropriate wire format and delivers it to the requesting client.

[0048Q]   Otherwise, in step 1204, if the requested resource is not available from the adaptable cache, the request is forwarded to host processor 120 for processing. In step 1205, host processor 120 formulates a request for the asset to storage system 102. In step 1206, the asset is returned to host processor 120, as described above in connection with Fig. 10. In step 1207, host

processor 120 converts the asset to an appropriate wire format and delivers it to the client via network interface 130.

[0048R] It should be recognized that the proactive caching and notification described above may also be implemented in this embodiment. Thus, adaptable cache 600 may be adapted to monitor received requests, proactively cache some or all of an asset in accordance with caching rules, and notify one or more applications or processes of content that it is currently storing. Further, the adaptable cache may be adapted to direct the storage system not to respond to requests for particular assets when the assets are cached in the adaptable cache.

[0048S] Another preferred embodiment for implementing the present system and method is shown in Fig. 13. In the embodiment of Fig. 13, adaptable cache 600 is integrated with controller 128, bridging I/O buses 106A, B. In this embodiment, adaptable cache 600 preferably plugs into an expansion slot on the host system and provides multiple standard high-speed interfaces, such as bridging Fibre Channel and PCI I/O interfaces. In this embodiment, preferred physical specifications of the adaptable cache include:

- the form factor of a peripheral component interconnect (PCI) card;
- storage capacity in excess of 1 gigabyte (GB) using replaceable commercially off-the-shelf memory modules (such as dual inline memory modules - DIMMs) or fixed memory circuits; and
- conformity to PC1 hot-swap specifications to allow the adaptable cache to be removed from service while the host system is in operation. As noted above, the

FACSIMILE

storage size of the adaptable cache can therefore be altered through a hot-swap without disrupting the operation of the media server.

[0048T]   Operation of the system shown in Fig. 13 will now be described in connection with Fig. 14. In step 1401, a user request is received at network interface 130. In step 1402, the request is forwarded to host processor 120 via I10 bus 106B. In step 1403, host processor 120 sends a request for the asset to storage system 102 via I/O bus 106B.

[0048U]   In step 1404, adaptable cache 600 (integrated with controller 128 in this embodiment) monitors asset requests that traverse I/O buses 106A, B and determines if the requested asset is available on the adaptable cache. In step 1405, if the asset is available on the adaptable cache, it is returned to host processor 120.

[0048V]   Otherwise, if the requested resource is unavailable from the adaptable cache, the request is forwarded to storage system I/O bus 106A for delivery to the appropriate storage device 104 where the resource persists (step 1406). In step 1407, the storage device, returns the resource to the requesting application, as described in more detail above. In step 1408, host processor 120 receives the requested resource, as described in more detail above.

[0048W]   It should be recognized that the proactive caching and notification described above may also be implemented in this embodiment. Thus, adaptable cache 600 may be adapted to monitor received requests, proactively cache some or all of an asset in accordance with caching rules, and notify one or more applications or processes of content that it is currently storing.

FACSIMILE

Further, the adaptable cache may be adapted to direct the storage system not to respond to requests for particular assets when the assets are cached in the adaptable cache.

[0048X] Yet another preferred embodiment for implementing the present system and method is shown in Fig. 15. In Fig. 15, an adaptable cache 600 resides on storage system 102. Adaptable cache 600 preferably interconnects with storage system I/O bus 106A via a high-speed interface connection 1500. This high-speed interface connection preferably allows adaptable cache 600 to replace or supplement existing hard drive storage devices on storage system 102 (including RAID arrays or JBODS) as long as the system has a compatible receptacle and I/O interface. In this embodiment, preferred physical specifications of the adaptable cache comprise:

- the form factor of a 3.5" hard disk drive with a 1" nominal height;
- dual fibre channel interface utilizing a standard SCA 40-pin connector and operating at transfer rates of either 1 or 2 gigabits per second (Gbps);
- storage capacity in excess of 1 gigabyte (GB) using replaceable commercially off-the-shelf memory modules (such as dual inline memory modules - DIMMs) or fixed memory circuits - this facilitates a lower cost while simultaneously providing the benefit of readily available and quality controlled components; and
- hot-swap capability (the ability to swap or remove the adaptable cache from service while the system is in operation). As noted above, the storage size of the adaptable cache can therefore be altered through a hot-swap without disrupting the operation of the media server.

FACSIMILE

[0048Y] Operation of the preferred embodiment shown in Fig. 15 will now be described in connection with Fig. 16. In step 1601, a user request is received at network interface 130. In step 1602, the request is forwarded to host processor 120 via I/O bus 106B. In step 1603, host processor 120 sends a request for the asset to storage system 102 via I/O bus 106B.

[0048Z] In step 1604, adaptable cache 600 monitors asset requests that traverse I/O bus 106A and determines if the requested asset is available on the adaptable cache. As noted above, those skilled in the art will recognize that passive monitoring of bus 106B by adaptable cache 600 may be impractical with more modern busses which are often segmented and behave more like networks in which each device sees only traffic specifically addressed to it. Accordingly, as noted above, in systems comprising such busses, host processor 120 may be adapted to address each received asset request to both storage device 104 and to adaptable cache 600 so that adaptable cache 600 may monitor traffic between host processor 120 and storage device 104.

[0048AA] In step 1605, if the asset is available on the adaptable cache, it is returned to host processor 120. In this case, the adaptable cache or other suitable component in storage system 102 may also preferably be adapted to preclude other storage devices 104 from responding to the request from host processor 120 since such storage device will be unable to retrieve and forward the asset to host processor 120 as efficiently as adaptable cache 600 (step 1606).

[0048AB] Otherwise, if the requested resource is unavailable from the adaptable cache, the request is delivered to the appropriate storage device 104 where the resource persists (step 1607).

FACSIMILE

In step 1608, the storage device returns the resource to the requesting application, as described in more detail above.

[0048AC]   It should be recognized that the proactive caching and notification described above may also be implemented in this embodiment. Thus, adaptable cache 600 may be adapted to monitor received requests transmitted via 110 bus 106A, proactively cache some or all of an asset in accordance with caching rules, and notify one or more applications or processes of content that it is currently storing. Alternatively, these caching and monitoring components may be divided. More specifically, a separate monitoring component may be provided on I/O bus 106A to monitor requests as they are received by network interface 130. When appropriate, the monitoring component may instruct adaptable cache 600 (residing, for example, on 110 bus 106A) to retrieve and store some or all of an asset.

[0048AD]   It should also be noted that although, in the preferred embodiments described above, system components are linked via PC1 buses such as bus 106A, B, these components may alternatively be linked via other bus types or data exchanges such as switched fabric and associated daughtercards.

FACSIMILE

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A method for selecting a server from a plurality of servers to service a request for requested content, comprising:

detecting the addition of new content to a first server's adaptable cache on a first server in the plurality of servers;

updating a first state table on the first server with information about the new content stored on the first server's adaptable cache;

communicating the information about the new content stored on the first server's adaptable cache to each server in the plurality of servers;

updating state tables of each of the other servers in the plurality of servers with the information about the new content stored on the first server's adaptable cache;

designating a director from the plurality of servers to receive the request, ~~wherein the designation is made on a request-by-request basis and~~ wherein any of the plurality of servers can be designated as the director;

determining that the requested content is not stored on ~~the director~~ a director's adaptable cache by accessing a director's state table stored on the director, wherein the director's state table includes parametric information for each server in the plurality of servers, and wherein the parametric information

comprises adaptable cache contents information for each server in the plurality of
servers; and

under the direction of the director,

determining ~~whether any other~~ a set of servers from among said
plurality of servers that have ~~has~~ the requested content stored ~~thereon~~ in
their respective adaptable caches by examining the state table on the
director;

determining a load factor for each of the ~~other servers having the
requested content~~ set of servers; and

selecting ~~one of the other servers having the requested content to
service the request,~~ a second server from among the set of servers based
on the load factor.

2.   (Original) The method of claim 1, wherein the step of designating
comprises designating the director in a round-robin fashion.

3.   (Previously presented) The method of claim 1, wherein the director is
designated based on a load factor analysis for each server among said plurality of
servers, the load factor for each server based on parametric information stored in a
respective state table thereon, and wherein the designated director has a lowest load
factor.

FACSIMILE

4.   (Previously presented) The method of claim 1, further comprising selecting the director upon determining that the requested content is present on the director.

5.   (Currently amended) The method of claim 1, wherein said parametric information ~~further~~ comprises functional state and current load of each server.

6.   (Currently amended) The method of claim 1, wherein said parametric information ~~further~~ comprises whether each server comprises extended memory.

7.   (Currently amended) The method of claim 1, wherein said parametric information ~~further~~ comprises whether each server comprises an inline adaptable cache.

8.   (Currently amended) The method of claim 1, wherein said parametric information ~~further~~ comprises whether each asset represented in the parametric information is a new release.

9.   (Currently amended) The method of claim 1, further comprising ~~rejecting the request upon determining that the requested content is not present on any of the plurality of servers~~ storing the requested content on the director's adaptable cache responsive to the request.

10. (Currently amended) The method of claim 1, further comprising forwarding the request to the ~~selected~~ second server.

11. (Currently amended) The method of claim 1, further comprising redirecting the request to the ~~selected~~ second server.

12. (Currently amended) The method of claim 1, wherein ~~the step of~~ selecting the second server from among the set of servers further comprises:

identifying as available servers any servers whose load factors are below threshold limits;

determining [[if]] that there are [[any]] no available servers; and

upon determining that there are no available servers, selecting a server having a lowest load factor from the other servers having the content.

13. (Currently amended) A server computer configured to direct a request for content among a plurality of server computers comprising:

a state table comprising parametric information for each server in the plurality of server computers, said state table enabling any one of the plurality of server computers to act as a director, said parametric information comprising information identifying assets maintained in adaptable caches on the plurality of server computers; and

FACSIMILE

a communication component for concurrently pushing changes to the state table to each of the other servers in said plurality of server computers upon any such change, wherein the addition of an asset to <u>an adaptable cache of </u>the server computer initiates a change to the state table of the server computer <u>indicating the addition of the asset to the adaptable cache of the server computer</u> and a transmission of information about the ~~change~~ <u>the addition of the asset to the adaptable cache of the server computer </u>to each of the other servers in said plurality of server computers.

14.   (Previously Presented)   The server of claim 13, wherein the server computer is a member of a load-balancing group, and the communication component sends changes to server computers in the load-balancing group.

15.   (Previously Presented) The server of claim 13, further comprising a redirection means for identifying one of the plurality of server computers where a requested asset is stored.

16.   (Previously Presented) The server of claim 13, further comprising a forwarding means for sending the request to one of the plurality of server computers where a requested asset is stored.

FACSIMILE

17. (Previously Presented) The server of claim 13, wherein said parametric information further comprises functional state and current load of each server computer.

18. (Previously Presented) The server of claim 13, wherein said parametric information further comprises whether each server computer comprises extended memory.

19. (Previously Presented) The server of claim 13, wherein said parametric information further comprises whether each server computer comprises an inline adaptable cache.

20. (Original) The server of claim 13, wherein said parametric information further comprises whether each asset is a new release.

21. (Currently amended) A computer-readable medium comprising computer-executable instructions for performing a method comprising:

adding new content to a first server's adaptable cache on a first server in a plurality of servers, wherein the first server updates a first state table on the first server with information about the new content stored on the first server's adaptable cache, wherein the first server ~~communicate~~ communicates the information about the new content stored on the first server's adaptable cache to each server in the plurality of servers, and wherein each server in the plurality of

FACSIMILE

servers updates each state table of each server in the plurality of servers with the information about the new content stored on the first server's adaptable cache;

designating a director from the plurality of servers to receive the request, ~~wherein the designation is made on a request-by-request basis and~~ wherein any of the plurality of servers can be designated as the director;

determining that the requested content is not stored on ~~the director~~ a director's adaptable cache by accessing a state table stored on the director, wherein the state table includes parametric information for each server in the plurality of servers, and wherein the parametric information comprises adaptable cache contents information for each server in the plurality of servers; and

under the direction of the director,

determining ~~whether any other~~ a set of servers from among said plurality of servers that have ~~has~~ the requested content stored ~~thereon~~ in their respective adaptable caches by examining the state table on the director;

determining a load factor for each of the ~~other servers having the requested content~~ set of servers; and,

selecting ~~one of the other servers having the requested content to service the request,~~ a second server from among the set of servers based on the load factor.

FACSIMILE

22.　(Previously presented)　The computer-readable medium of claim 21, wherein the step of designating comprises designating the director in a round-robin fashion.

23.　(Previously presented)　The computer-readable medium of claim 21, wherein the step of designating comprises designating the director on the basis of lowest load.

24.　(Previously presented)　The computer-readable medium of claim 21, wherein the step of selecting further comprises selecting the director if the requested content is present on the director.

25.　(Currently amended)　The computer-readable medium of claim 21, wherein said parametric information _further_ comprises functional state and current load of each server.

26.　(Currently amended)　The computer-readable medium of claim 21, wherein said parametric information _further_ comprises whether each server comprises extended memory.

FACSIMILE

27.   (Currently amended) The computer-readable medium of claim 21, wherein said parametric information <u>further </u>comprises whether each server comprises an inline adaptable cache.

28.   (Currently amended) The computer-readable medium of claim 21, wherein said parametric information <u>further </u>comprises whether each asset represented in the parametric information is a new release.

29.   (Currently amended) The computer-readable medium of claim 21, further comprising computer-executable instructions for ~~rejecting the request if the requested content is not present on any of the plurality of servers~~ <u>storing the requested content on the director's adaptable cache responsive to the request</u>.

30.   (Currently amended) The computer-readable medium of claim 21, further comprising computer-executable instructions for forwarding the request to the ~~selected~~ <u>second </u>server.

31.   (Currently amended) The computer-readable medium of claim 21, further comprising computer-executable instructions for redirecting the request to the ~~selected~~ <u>second </u>server.

FACSIMILE

32.　(Currently amended)　The computer-readable medium of claim 21, wherein the step of selecting <u>the second server from among the set of servers </u>further comprises:

identifying as available servers one or more servers whose load factors are below threshold limits;

determining that there are no available servers; and

upon determining that there are no available servers, selecting a server having a lowest load factor from the other servers having the content.

33.　(Previously Presented)　The method of claim 1, further comprising updating parametric information in a state table associated with the selected server, and communicating updated parametric information to the other servers among said plurality of servers.

34.　(Previously Presented)　The method of claim 33, wherein the updated parametric information is communicated via multicast.

35.　(Previously Presented)　The method of claim 33, wherein the updated parametric information is communicated via a broadcast message.

FACSIMILE

Amendments to the Drawings:

Please replace the originally filed drawings with the attached sheets of drawings include Figures 1-16. These sheets include Figures 1-8 as originally filed and Figures 9-16 that are described in the amendments to the specification recited above. These sheets replace the original sheets including Figures 1-8.

Attachment: Replacement Sheets